# (NeXT Tip #25a) Text Object Utilities

Christopher Lane (*lane[at]CAMIS.Stanford.EDU*)
*Tue, 23 Mar 1993 15:18:02 -0800 (PST)*

When you drag a Text object from the InterfaceBuilder palette into your application, there are two basic methods to write into it: 'setText:' which lets you set it's contents to a string and 'readText:' which lets you fill the Text object from a file (using an open NXStream). They're also 'readRichText:' and 'readRTFDFrom:' which are RTF variants of 'readText:'.

However, these both assume you want to completely replace the contents of the view. Just adding more text to takes a little more work and requires using 'replaceSel:' or one of its variants. Below are two example methods that let you add text incrementally to the end of a Text-based ScrollView.

The first, 'appendText:', is like 'readText:' except that it adds text from a stream to the end of a Text-based ScrollView and is a trivial method.

The second, 'printf:', which I originally posted some years ago, is a little more complex and gives you the full power of 'printf', variable argument count and all, when writing to a Text object.

Since these two methods don't have application specific features, I've cast them as 'categories' -- additional methods for the Text object itself. They could just as easily be implemented as methods of a subclass of Text or other ways. In my own usage, I typically have the 'printf:' method return 'nil' instead of 'self' so that it can be used as an error return:

else return [messageView printf:"File not found: %s\n", syntaxnamebuf];

Both of these methods can probably be upgraded to RTF by using a memory-based scratch NXStream instead of the 'char' buffer, among other changes. An exercise left to the reader!

- Christopher

```
#import <appkit/Text.h>

@interface Text(PatchMethods)

- appendText:(NXStream *) stream;
- printf:(const char *) format, ...;

@end

@implementation Text(PatchMethods)

- appendText:(NXStream *) stream
{
    int count, length;
    char buffer[BUFSIZ + 1];

    while((count = NXRead(stream, (void *) buffer, BUFSIZ)) > 0) {
        if(count > 0) {
            buffer[count] = '\0';
            length = [self textLength];
            [[self setSel:length :length] replaceSel:buffer];
            }
        if(count < BUFSIZ) break;
        }

    return [[self scrollSelToVisible] display];
}

- printf:(const char *) format, ...
{
    va_list ap;
    char buffer[BUFSIZ];
    int length = [self textLength];

    va_start(ap, format); {
        (void) vsprintf(buffer, format, ap);
        } va_end(ap);

    [[self setSel:length :length] replaceSel:buffer];

    return [[self scrollSelToVisible] display];
}

@end
```